Flot de scène

Antoine Letouzey, Benjamin Petit, Edmond Boyer

Flot de scène

Antoine Letouzey, Benjamin Petit, Edmond Boyer

INRIA Grenoble Rhône-Alpes
655, Avenue de l'Europe 38334, S^t Ismier, France
prenom.nom@inria.fr

RÉSUMÉ. *Dans cet article nous nous intéressons à l'estimation des champs de déplacement 3D denses d'une scène non rigide, en mouvement, capturée par un système multi-caméra. La motivation vient des applications multi-caméra qui nécessitent une information de mouvement pour accomplir des tâches telles que le suivi de surface ou la segmentation. Dans cette optique nous présentons une approche nouvelle qui permet de calculer efficacement un champ de déplacement 3D, en utilisant des informations visuelles de bas niveau et des contraintes géométriques. La contribution principale est la proposition d'un cadre unifié qui combine des contraintes de flot pour de petits déplacements et des correspondances temporelles éparses pour les déplacements importants. Ces deux types d'informations sont fusionnés sur une représentation surfacique de la scène en utilisant une contrainte de rigidité locale. Le problème se formule comme une optimisation linéaire permettant une implémentation rapide grâce à une approche variationnelle. La méthode proposée s'adapte de manière quasiment identique que les informations de surface proviennent d'une reconstruction 3D complète, par exemple en utilisant l'enveloppe visuelle, ou d'une simple carte de profondeur. Les expérimentations menées sur des données synthétiques et réelles démontrent les intérêts respectifs du flot et des informations éparses, ainsi que leur efficacité conjointe pour calculer les déplacements d'une scène dynamique de manière robuste. Cet article est une version étendu de l'article (Letouzey* et al.*, 2012) présenté à RFIA 2012.*

ABSTRACT. *In this paper we consider the problem of estimating a 3D motion field using multiple cameras. In particular, we focus on the situation where a depth camera and one or more color cameras are available, a common situation with recent composite sensors such as the Kinect. In this case, geometric information from depth maps can be combined with intensity variations in color images in order to estimate smooth and dense 3D motion fields. We propose a unified framework for this purpose, that can handle both arbitrary large motions and sub-pixel displacements. The estimation is cast as a linear optimization problem that can be solved very efficiently. The novelty with respect to existing scene flow approaches is that it takes advantage of the geometric information provided by the depth camera to define a surface domain over which photometric constraints can be consistently integrated in 3D. Experiments on real and synthetic data provide both qualitative and quantitative results that demonstrate the interest of the approach.*

Problem Formulation *In order to estimate the 3D flow, we cast the problem as an minimization where data terms corresponding to photometric consistency constraints are combined with a regularization term that favors smooth motion fields :*

$$\mathbf{E} = \mathbf{E}_{data} + \mathbf{E}_{smooth}. \tag{1}$$

Data terms enforce visual coherence of the computed displacement field while the regularization term imposes a deformation model with local rigidity constraints.

Visual Constraints *As suggested by Xu* et al. *in their work on optical flow (Xu* et al.*, 2010), we use two different kinds of photometric cues to deal with both large and small displacements. First we match sparse* visual features *(SIFT) between two consecutive color images. This information is not sensitive to the amplitude of the motion in the scene. For small details we use the well known* normal flow *information available at every pixels but only valid for small motion. Both cues contribute a term to* $\mathbf{E}_{data}$ *in equation 1.*

Geometric Constraints *The regularization stage is important for two reasons. First we need to propagate the sparse cues given by the visual features, and second the* aperture problem*, well known in optical flow estimation, extends from 2D to 3D. Hence the data term in our formulation is not sufficient to compute the scene flow. Unlike existing work (Vedula* et al.*, 2005), we choose to perform this regularization using 3D information given by surface, instead of computing optical flow in the image domain and do a projection of this 2D flow on the depth maps. We extended Horn & Schunck's method (Horn, Schunck, 1981) to 3D. This regularization enforces a global smoothness of the motion field in 3D. Therefore we do not suffer from 2D regularization-specific drawbacks, such as object boundaries and depth discontinuities oversmoothing. This geometric constraint yields the smoothing term in equation 1.*

Formulation & Resolution *We gather all the visual and geometric constraints into one single linear system of the following form :*

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{A} \end{bmatrix} V + \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} = 0\,, \tag{2}$$

where $\mathbf{L}$ *is the Laplacian matrix of the mesh associated to 3D surface,* V *is a vector compounding the motion of all the scene points, and* $\mathbf{A}$ *and* $\mathbf{b}$ *stack all the motion constraints coming from data terms. The paper explains in details the construction of these matrices along with a discussion about Laplacian weights. This linear system is very sparse.*
In practice, we propose a two-step algorithm. The first one handles large displacements, and the second recovers small motion details. This is done by adjusting the weight associated to each constraint and perform two consecutive resolutions of equation 2.

Results *We tested our approach on both synthetic and real data. Figure 14, for instance, shows some results on real data. We used different setups with either one or two color cameras and a depth camera or a multi-camera setups with up to 32 color cameras. We also tested two different depth camera types, a* time-of-flight *camera and a Kinect camera. Synthetic data allowed us to perform a numerical comparison between our method and the one proposed in (Vedula* et al.*, 2005).*

Contribution *(i) Following works on robust optical flow estimation (Xu* et al.*, 2010), we take advantage of robust initial displacement values as provided by image features tracked over consecutive time instants. (ii) A linear framework that combines visual constraints with surface deformation constraints and allows for iterative resolution (variational approach) as well as coarse-to-fine refinement.*

MOTS-CLÉS : Déplacement 3D, flot de scène, carte de profondeur, surface

KEYWORDS: 3D motion, Scene Flow, Depth map, Surface

DOI:10.3166/TS.22.1-28

1. Contexte et motivations

FIGURE 1 – Exemple de flot de scène dense (en bleu) calculé à partir de correspondances de points d'intérêts 2D et 3D et de flot de normal dense.

Le déplacement est une source d'information importante lors de l'analyse et de l'interprétation de scènes dynamiques. Il fournit une information riche et discriminante sur les objets qui composent la scène et est utilisé, par exemple, dans les systèmes de vision humaine et artificielle pour suivre et délimiter ces objets. L'intérêt apparaît surtout dans le cas d'applications interactives, telles que les jeux vidéos ou les environnements intelligents, pour lesquels le mouvement est une source d'information primordiale dans la boucle perception-action. Pour cela, l'observation des pixels, issus des images, fournit des informations utiles sur le mouvement, à travers les variations temporelles de la fonction d'intensité. Dans une configuration mono-caméra, ces variations permettent d'estimer des champs de vitesse 2D denses dans l'image : le *flot optique*. L'estimation du flot optique a été un sujet d'intérêt dans la communauté de la vision par ordinateur ces dernières dizaines d'années et de multiples méthodes ont été proposées (Barron *et al.*, 1994 ; Horn, Schunck, 1981 ; Lucas, Kanade, 1981).

Dans le cas d'un système multi-caméra, l'intégration depuis les différents points de vue permet de considérer le mouvement des points 3D de la surface observée et d'estimer le champ de vecteur de déplacement 3D : le *flot de scène* (Vedula *et al.*, 2005 ; Neumann, Aloimonos, 2002). Autant en 2D qu'en 3D, l'information de mouvement ne peut pas être déterminée indépendamment pour chaque point avec pour seule information la variation de la fonction d'intensité ; une contrainte additionnelle doit-être introduite, par exemple, une hypothèse de continuité du champ de mouvement. De plus, du fait de l'approximation des dérivées par la méthode des différences finies, l'estimation du flot est connue pour être limitée à de petits déplacements. Bien que plusieurs approches en 2D aient été proposées pour faire face à ces limitations (Xu *et al.*, 2010), moins d'efforts ont été consacrés au cas de la 3D. Il est bien sur possible d'utiliser des capteurs actifs ou des systèmes de vision basés marqueurs. Ces derniers peuvent fournir directement un ensemble épars d'informations de déplacement sur des scènes en mouvement. Mais ces systèmes sortent du cadre de nos travaux, en effet nous nous contraignons à utiliser un système le moins intrusif possible, c'est-à-dire sans marqueur et sans hypothèse sur l'éclairage, voir même sous éclairage naturel.

Dans ce travail, nous avons étudié la façon d'intégrer, de manière efficace, diverses contraintes pour estimer des informations de mouvements denses instantanés sur des surfaces 3D, à partir des variations temporelles de la fonction d'intensité issue de plusieurs images. Notre motivation première a été de fournir des indices de mouvement robuste qui peuvent être directement utilisés par une application interactive, ou qui peuvent être introduits dans des applications plus avancées comme le suivi de surface ou la segmentation. Bien que notre but ait été d'intégrer le calcul des champs de vitesse avec notre application de reconstruction 3D, l'approche n'est pas limitée à un scénario spécifique et fonctionne pour toute application qui peut bénéficier d'une information de mouvement de bas niveau.

La plupart des approches existantes qui estiment le flot de scène font l'hypothèse des petits déplacements entre les instants de temps pour lesquels les approximations aux différences finies des dérivées temporelles sont valides. Cependant, cette hypothèse est souvent incorrecte avec les systèmes d'acquisition actuels et des objets réels en mouvement. En effet, l'amplitude des mouvements observés et la fréquence d'acquisition utilisée ne permettent pas d'effectuer cette hypothèse dans tous les cas.

Dans cet article, nous présentons une méthode unifiée permettant de lier de manière cohérente les contraintes visuelles, issues des images consécutives temporellement, avec des contraintes de déformation de surface. Pour traiter les grands déplacements, nous utilisons des mises en correspondances temporelles entre les images issues d'une même caméra. Ces contraintes agissent comme des points d'ancrage pour les régions de la surface où les déplacements sont plus importants et où les informations de variation d'intensité ne sont pas utiles. Ces contraintes visuelles sont diffusées sur la surface grâce à un schéma laplacien qui régularise les vecteurs de déplacements estimés entre les points voisins de la surface. Un élément clé de cette méthode est qu'elle conduit à des optimisations linéaires ce qui permettrait, à terme, une implémentation temps-réel.

Notion de Surface. La méthode que nous proposons ici nécessite en entrée un ensemble de flux d'images couleur venant d'un système multi-caméra pré-étalonné et nous supposons qu'une information géométrique sur la scène est disponible. Nous avons adapté notre méthode à deux cas de figures distincts. Le premier est le cas où l'information de surface provient d'une reconstruction 3D indépendante de chaque trame de la séquence traitée, en utilisant l'enveloppe visuelle par exemple. Le second cas de figure est un système où la géométrie partielle de la scène est donnée par une caméra de profondeur, par exemple des caméras à temps de vol ou à lumière structurée, qui fournissent directement une information 3D, sans recourir à un traitement multi-vue additionnel. La carte de profondeur représente un nuage de points 3D à partir duquel une surface maillée peut être construite en utilisant la connectivité dans l'image de profondeur. C'est-à-dire que chaque pixel devient un sommet du maillage en 3D, connecté à ses voisins dans l'image. Dans la suite de cet article, sauf mention contraire, nous appellerons *surface* cette information géométrique indistinctement de sa provenance.

Cet article est organisé de la manière suivante : dans un premier temps, nous présentons un état de l'art dans la section 2, ensuite nous entrons dans les détails de la méthode proposée dans la section 3. Dans la section 5, nous expliquons les différents choix d'implémentations que nous avons fait suivant le type de données traitées. Les résultats obtenus dans le cas des enveloppes visuelles et celui des cartes de profondeurs sont présentés respectivement dans les sections 6 et 7. Nous concluons cet article dans la section 8.

2. Etat de l'art

Un grand nombre de travaux ont été menés dans le but d'estimer des champs de déplacement en utilisant des informations photométriques. Les premiers travaux dans ce domaine se concentraient sur le champ de déplacement entre deux images consécutives. L'estimation du flot optique par (Horn, Schunck, 1981 ; Barron *et al.*, 1994) fait appel aux contraintes de flot normal dérivées des variations d'intensité dans les images. Lorsque l'information vient d'images stéréo, le champ de déplacement 3D, le flot de scène, peut être calculé.

Dans un article fondateur sur le flot de scène, Vedula *et al*. (Vedula *et al.*, 2005) explicitent la contrainte de flot normal qui lie les dérivées de la fonction d'intensité dans les images au flot de scène des points 3D de la surface. Comme mentionné précédemment, ces contraintes ne permettent pas d'estimer le flot de scène de façon indépendante à un point de la surface, des contraintes supplémentaires doivent être introduites. Au lieu d'utiliser la contrainte de flot normal, un algorithme est proposé qui estime de façon linéaire le flot de scène à partir de la géométrie 3D de la surface et du flot optique 2D. Le flot optique permet de mieux contraindre le flot de scène que le flot normal, mais son estimation est fondée sur des hypothèses de lissage qui tiennent rarement dans le plan image mais sont souvent vérifiées dans le cas de surfaces.

Dans (Neumann, Aloimonos, 2002), Neumann et Aloimonos introduisent un modèle de subdivision de surface qui permet d'intégrer sur la surface, les contraintes de flot normal avec des contraintes de régularisation. Néanmoins, cette solution globale suppose encore de n'être en présence que de petits mouvements et peut difficilement faire face à des cas comme ceux présentés dans nos expérimentations.

Une autre stratégie est d'estimer conjointement la structure et le mouvement. Cette voie est explorée par (Pons *et al.*, 2005 ; Basha *et al.*, 2010). Dans (Pons *et al.*, 2005) Pons *et al*., présentent une approche variationnelle qui optimise un critère de cohérence photométrique au lieu des contraintes de flot normal. L'intérêt est que la cohérence spatiale comme la cohérence temporelle peuvent être appliquées, mais au prix d'une optimisation coûteuse en calcul. Au contraire, notre objectif n'est pas d'optimiser la forme observée, mais de fournir une information de mouvement dense de façon efficace et rapide.

Plusieurs travaux (Zhang, Kambhamettu, 2001 ; Isard, MacCormick, 2006 ; Wedel *et al.*, 2008 ; Huguet, Devernay, 2007 ; Li, Sclaroff, 2008) considèrent le cas où la

structure de la scène est décrite par une carte de disparité issue d'un système stéréoscopique. Ils proposent l'estimation combinée de la disparité spatiale et temporelle du mouvement 3D. Des travaux récents (Rabe *et al.*, 2010) ajoutent à ceci une contrainte de cohérence temporelle. Nous considérons une situation différente dans laquelle la surface de la forme observée est connue, par exemple, un maillage obtenu en utilisant une approche multi-vues. Ceci permet une régularisation du champ de déplacement sur un domaine où les hypothèses de régularité sont vérifiées.

Il convient de mentionner également les approches récentes sur le suivi temporel de surface (Starck, Hilton, 2007b ; Varanasi *et al.*, 2008 ; Naveed *et al.*, 2008 ; Cagniart *et al.*, 2010) qui peuvent également fournir des champs de vitesse. C'est en effet une conséquence de la mise en correspondance de surfaces dans le temps. Notre but est ici différent, notre méthode ne fait aucune hypothèse sur la forme observée, seulement quelques hypothèses sur le modèle de déformation locale de la surface. Notre méthode fournit des informations bas niveau, le mouvement instantané, qui peuvent à leur tour être utilisées comme données d'entrée d'une méthode d'appariement ou de suivi de surface.

Nos contributions à l'égard des approches mentionnées sont de trois ordres :

– En suivant les travaux sur l'estimation robuste du flot optique 2D (Liu *et al.*, 2008 ; Xu *et al.*, 2010), nous utilisons avantageusement les valeurs de déplacement robuste fournies par le suivi de points d'intérêts dans des images consécutives dans le temps. Ces points intérêts permettent de contraindre les grands déplacements alors que les contraintes de flot de normal permettent de modéliser précisément les déplacements les plus petits.

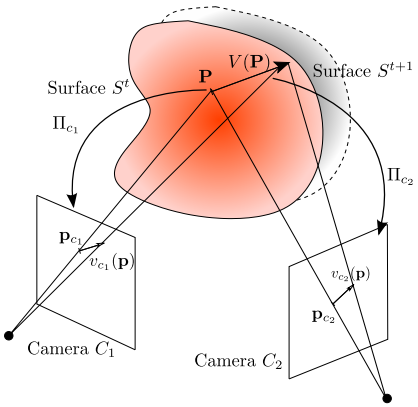
– Une résolution linéaire combine ces différentes contraintes visuelles avec un modèle de déformation de surface et permet une résolution itérative ainsi qu'un raffinement de type multi-échelle.

– Un cadre de résolution qui prend en compte les données venant de systèmes multi-caméra de natures différentes, contenant un nombre quelconque de caméras couleur et pouvant intégrer un capteur de profondeur.

3. Estimation du flot de scène

L'approche proposée estime directement un champ de mouvement 3D sur la surface en utilisant des contraintes photométriques 2D. Pour cela, elle prend en entrée des flux d'images couleur et de surfaces venant d'un système multi-caméra pré-étalonnées et synchronisées. La configuration prise en compte se compose d'une ou plusieurs caméras couleur et d'un flux de surface. Dans la suite, pour des raisons de simplicité, nous ne détaillons que le cas disposant d'une caméra couleur. Néanmoins l'extension à plusieurs caméras couleur est directe et sera expliquée plus loin dans cet article. Un des avantages de la méthode proposée est qu'elle s'adapte indifférement à une grande varité de systèmes d'acquisition. Du simple couple comprenant une caméra couleur et une caméra de profondeur, telle que la caméra Kinect, jusqu'à la salle d'acquisition complète contenant 32 caméras voir plus.

3.1. Notations



(a) Modèle du système multi-caméra

(b) Données multi-vue

(c) Données caméra de profondeur

FIGURE 2 – (a) Modèle multi-caméra considéré par notre approche, (b) type de données en entrée dans le cas où la surface est reconstruite par une méthode type enveloppe visuelle et (c) type de données dans le cas où l'on dispose d'une caméra de profondeur.

La surface au temps t est dénotée $\mathcal{S}^t \subset \mathbb{R}^3$ et associée à un ensemble d'images couleur, acquises au même instant de temps, noté $\mathcal{I}^t = \{\mathbf{I}_c^t \mid c \in [1..N]\}$. Un point 3D $\mathbf{P}$ sur la surface est décrit par le vecteur $(x, y, z)^T \in \mathbb{R}^3$. Sa projection dans l'image $\mathbf{I}^t$ est le point 2D $\mathbf{p}$ qui a comme coordonnées $(u, v)^T \in \mathbb{R}^2$, calculées en utilisant la matrice de projection 3x4 $\mathbf{\Pi} : \mathbb{R}^3 \mapsto \mathbb{R}^2$ de la caméra (voir figure 2). La région 3D de l'image correspondant à la visibilité de $\mathcal{S}^t$ dans $\mathbf{I}^t$ est notée $\Omega^t = \mathbf{\Pi}\mathcal{S}^t$.

Notre méthode recherche le meilleur champ de déplacement 3D de la surface entre le temps t et $t+1$, noté $V^t : \mathcal{S}^t \mapsto \mathbb{R}^3$ avec $V^t(\mathbf{P}) = \frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t}\ \forall\ \mathbf{P} \in \mathcal{S}^t$. Ce champ de déplacement est contraint par :

- les données d'entrée comme le jeu d'images calibrées $\mathbf{I}^t$ et $\mathbf{I}^{t+1}$, et les surfaces $\mathcal{S}^t$ et $\mathcal{S}^{t+1}$,
- un modèle de déformation.

Ainsi le flot optique v^t est la projection du champ du flot de scène V^t sur l'image couleur $\mathbf{I}^t$. La relation entre un petit déplacement à la surface de $\mathcal{S}^t$ et son image prise par la caméra couleur est décrite par la matrice jacobienne 2x3 $J_{\mathbf{\Pi}}(\mathbf{p}) = \frac{\partial \mathbf{p}}{\partial \mathbf{P}}$, telle que $v^t = J_{\mathbf{\Pi}}(\mathbf{p})V^t$. Pour estimer le flot 3D $V^t(\mathbf{P})$, le problème est formulé sous la forme d'une optimisation où un terme d'attache aux données renforçant les contraintes photométriques est associé à un terme de lissage favorisant un champ de déplacement régulier :

$$\mathbf{E} = \mathbf{E}_{data} + \mathbf{E}_{smooth}. \tag{3}$$

Le terme d'attache aux données contrôle à la fois les grands et petits déplacements tandis que le terme de lissage impose un modèle de déformation avec des contraintes de rigidité locale.

Dans les sections suivantes, nous expliciterons les contraintes visuelles et géométriques venant des données en entrée et le modèle de déformation utilisé pour propager le mouvement sur la surface.

3.2. Contraintes visuelles

Notre méthode peut utiliser trois types de contraintes visuelles pour estimer le déplacement 3D :

1. des contraintes denses de flot normal dans les images,
2. des correspondances éparses de points d'intérêts 3D,
3. des correspondances éparses de points d'intérêts 2D.

Chacune de ces contraintes mènera à un terme dans notre fonctionnelle d'erreur telle qu'elle sera réécrite dans la section 4 et qui décrit comment le champ de déplacement estimé se rapporte aux observations. Ces contraintes n'incluent pas de cohérence photométrique spatiale ou temporelle car ces dernières impliquent des termes non linéaires dans la fonctionnelle d'erreur. Elles sont plus adaptées aux problèmes liés à l'optimisation de la forme de la surface qu'à l'estimation plus bas niveau du mouvement.

3.2.1. Flot normal dense

Des informations denses sur V^t peuvent être obtenues en utilisant le flot optique accessible dans les images. En effet, en prenant comme hypothèse que l'illumination reste constante entre $\mathbf{p}^{t+1}$ et $\mathbf{p}^t$, la projection du même point de la surface entre deux trames consécutives, on peut définir l'équation du **flot normal** (Barron *et al.*, 1994) comme étant :

$$\nabla I^t.v^t + \frac{\mathrm{d}I^t}{\mathrm{d}t} = 0\,,$$

ou équivalent en 3D à (Vedula *et al.*, 2005) :

$$\nabla I^t.\left[J_{\mathbf{\Pi}}V^t\right] + \frac{\mathrm{d}I^t}{\mathrm{d}t} = 0\,.$$

La fonction d'erreur suivante décrit comment la projection v^t dans l'image du champ de déplacement 3D calculé vérifie la contrainte de flot normal :

$$\mathbf{E}_{flow} = \int_{\Omega^t} \|\nabla I^t.\left[J_{\mathbf{\Pi}}V^t\right] + \frac{\mathrm{d}I^t}{\mathrm{d}t}\|^2\,\mathrm{d}\mathbf{p}\,. \tag{4}$$

Cependant, cette fonctionelle ne permet de contraindre le mouvement 2D que dans la direction tangente au gradient d'intensité dans l'image ∇I^t. C'est-à-dire que seule la projection du vecteur de flot optique sur l'axe du gradient d'intensité dans l'image est

connue. Cette limitation est connue sous le nom de problème de l'ouverture (*aperture problem*) dans le cas du l'estimation du flot optique. Il s'avère que ce problème s'étend en 3D pour l'estimation du flot de scène. En reprenant la démonstration faite par Vedula *et al*. dans (Vedula *et al.*, 2005), nous pouvons noter que les contraintes de flot normal ne sont pas dépendantes du point de vue et qu'ainsi, quelque soit le nombre de points de vue considérés, l'information accumulée sera toujours ambigüe. En effet la seule information complète qu'il est possible d'obtenir est la projection du flot de scène associé à un point $\mathbf{P}$ de la surface sur le plan tangent à la surface en ce même point $\mathbf{P}$.

3.2.2. Correspondances 3D éparses

Dans certaines situations, mouvements de faible intensité ou haute fréquence d'acquisition par exemple, le champ de déplacement peut être estimé uniquement à l'aide des contraintes denses de flot normal (accompagnées d'une régularisation). Néanmoins dans un contexte plus général, nous devons considérer d'autres sources d'information. La mise en correspondance de points d'intérêts 3D permet de recueillir de l'information pour un jeu de points 3D à la surface de $\mathcal{S}^t$. Ces points 3D et leurs déplacements associés sont obtenus par la détection des points d'intérêts 3D sur $\mathcal{S}^t$ et $\mathcal{S}^{t+1}$, en leur créant un descripteur et en les appariant grâce à la comparaison de ces descripteurs.

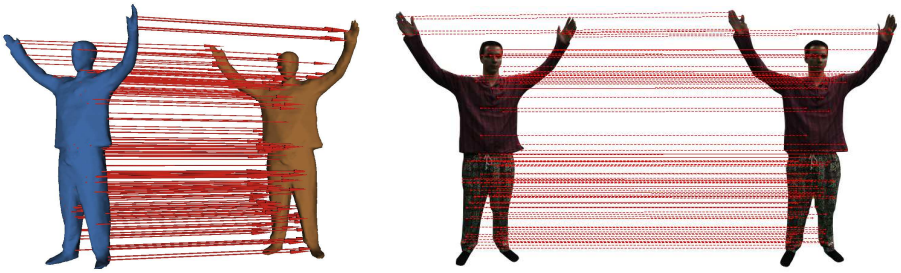
Il existe différentes voies pour obtenir des correspondances 3D entre deux formes. Dans notre approche, nous utilisons MeshDOG pour détecter des points d'intérêts 3D et MeshHOG pour les décrire (Zaharescu *et al.*, 2009). Cette méthode définit et met en correspondance les extremas locaux de n'importe quelle fonction scalaire définie sur la surface. Dans le cas où l'information géométrique provient d'une image de profondeur, des méthodes de détection et d'appariement de point d'intérêts 2D, tels que ceux décrits dans la section suivante, peuvent être utilisés directement sur celle ci. D'autres techniques de détection et mise en correspondance peuvent être utilisées (telle que (Starck, Hilton, 2007a)), tant qu'elles récupèrent un ensemble de correspondances robustes entre les deux surfaces. L'avantage de l'utilisation de points d'intérêts 3D est que, contrairement au flot optique (décrit à la section 3.2.1), ils permettent de contraindre le mouvement même lors de grands déplacements dans l'espace 3D.

On obtient un ensemble épars de déplacements 3D V_m^t pour des points 3D $\mathbf{P}_m \in \mathcal{S}^t$ (voir figure 3a). Ces points forment un sous-ensemble discret de $\mathcal{S}^t$ appelé $\mathcal{S}_m^t$. La fonction d'erreur suivante décrit la proximité du champ de déplacement calculé V^t au champ de déplacement épars V_m^t :

$$\mathbf{E}_{3D} = \sum_{\mathcal{S}_m^t} \|V^t - V_m^t\|^2 \ . \tag{5}$$

3.2.3. Correspondances 2D éparses

Dans notre approche, nous considérons des correspondances 2D éparses entre les images $\mathbf{I}^t$ et $\mathbf{I}^{t+1}$. Comme dans le cas de la 3D, il y a différentes techniques exis-



(a) Correspondances 3D–3D (b) Correspondances 2D–2D

FIGURE 3 – (a) Correspondances 3D éparses entre deux surfaces et (b) correspondances 2D éparses entre deux images.

tantes pour calculer des correspondances 2D entre une paire d'images, par exemple, SIFT (Lowe, 2004), SURF (Bay *et al.*, 2006) ou Harris (Harris, Stephens, 1988). Sans pour autant perdre en généralité, nous nous appuyons sur le détecteur et descripteur SIFT. Il s'est avéré robuste et bien adapté dans notre cas, car invariant aux rotations et aux changements d'échelle. Nous calculons des points d'intérêts sur les images $\mathbf{I}^t$ et $\mathbf{I}^{t+1}$. Nous mettons ensuite en correspondance les points d'intérêts ainsi obtenus. Cela nous donne un jeu de déplacements 2D épars v_s^t pour quelques points 2D $\mathbf{p}_s \in \Omega^t$ (voir figure 3b). Ces points forment un sous-ensemble de Ω^t appelé Ω_s^t. La fonction d'erreur suivante décrit la proximité du champ de déplacement 2D calculé v^t au champ de déplacement 2D épars v_s^t :

$$\mathbf{E}_{2D} = \sum_{\Omega_s^t} \|v^t - v_s^t\|^2 ,$$

ce qui est équivalent à :

$$\mathbf{E}_{2D} = \sum_{\Omega_s^t} \|J_{\mathbf{\Pi}} V^t - v_s^t\|^2 . \tag{6}$$

Il est important de noter que des correspondances 3D peuvent être obtenues à partir des points d'intérêts 2D en re-projetant les points détectés depuis $\mathcal{I}^t$ sur la surface $\mathcal{S}^t$ et ceux de $\mathcal{I}^{t+1}$ sur la surface $\mathcal{S}^{t+1}$. Cela fournit une liste de points d'intérêts 3D qui peuvent être mis en correspondance grâce à leurs descripteurs SIFT. Au lieu de réaliser cette mise en correspondance seulement dans l'espace d'une seule image, cela permet de prendre en compte les descripteurs issus de plusieurs images. Dans ce cas, la fonctionnelle d'erreur est similaire à $\mathbf{E}_{3D}$ décrite dans l'équation (5).

Bien que les points d'intérêts 3D soient plus robustes, en particulier aux occultations, et fournissent une meilleure information sur de longues séquences, ils présentent des désavantages, par rapport aux points d'intérêts 2D, pour l'estimation du flot de scène. Ils ne sont pas robustes aux changements de topologie et sont plus demandeurs en puissance de calcul, ce qui peut être crucial dans certaines applications.

De plus, la surface S^{t+1} est forcément requise ce qui peut être problématique pour des applications interactives.

3.3. Contrainte de régularité

Les correspondances éparses 2D et 3D contraignent seulement le déplacement de la surface pour des points 3D spécifiques et pour leur re-projection dans les images. Pour trouver un champ de mouvement dense sur la surface, nous avons besoin de propager ces contraintes en utilisant un terme de régularisation. En outre, comme mentionné précédemment, les contraintes denses de flot normal ne fournissent pas assez de contraintes pour estimer les déplacements 3D. En effet, il peut être démontré que les équations du flot normal pour des projections dans différentes images d'un même point 3D $\mathbf{P}$ contraignent de façon indépendante V^t à $\mathbf{P}$, et ne résolvent donc que 2 degrés de liberté sur 3. Vedula *et al*. (Vedula *et al.*, 2005) mentionnent deux stratégies de régularisation pour faire face à cette limitation. La régularisation peut être effectuée dans les plans images en estimant les flux optiques qui fournissent des contraintes plus complètes sur le flot de scène, ou elle peut être effectuée sur la surface 3D.

Puisque nous avons connaissance de la surface 3D et que les contraintes éparses 2D et 3D doivent être également intégrées, un choix naturel dans notre contexte est de régulariser en 3D. En plus, la régularisation dans l'espace image souffre d'artefacts et d'incohérences résultant des discontinuités de profondeur et des occultations qui contredisent l'hypothèse de lissage, alors qu'une telle hypothèse se justifie sur la surface 3D.

3.3.1. Modèle de déformation

Les hypothèses de régularité sur les champs de déplacement 3D de la surface limitent les déformations de cette surface à un niveau local. Elles définissent ainsi un modèle de déformation de la surface, par exemple, une rigidité locale. En 2D, de nombreuses méthodes de régularisation ont été proposées pour l'estimation du flot optique, elles se répartissent en 2 grandes catégories : les régularisations locales ou globales. Elles peuvent être étendues à la 3D. Par exemple, la méthode 2D de Lucas et Kanade, qui utilise un voisinage local, a été appliquée en 3D par Devernay *et al*. (Devernay *et al.*, 2006). Toutefois, le modèle de déformation associé à la surface n'a pas de signification réelle, car les contraintes de déformation ne se propagent que localement, ce qui amène à des incohérences entre les voisins. D'autre part, la stratégie globale introduite par Horn et Schunck (Horn, Schunck, 1981) est bien mieux adaptée à notre contexte. Bien que moins robuste au bruit que les méthodes locales telles que Lucas-Kanade, elle permet la propagation de contraintes éparses sur toute la surface. En outre, le modèle de déformation associé a prouvé son efficacité dans le domaine du graphisme (Sorkine, Alexa, 2007). L'extension du modèle de déformation d'Horn et

Schunck à des points 3D est décrit par la fonction d'erreur suivante qui assure une rigidité locale du champ de mouvement :

$$\mathbf{E}_{smooth} = \int_S \|\nabla V\|^2 \mathrm{d}\mathbf{P}\,. \tag{7}$$

4. Formulation et résolution

En regroupant tous les termes précédemment définis, notre fonctionelle d'énergie présentée dans l'équation (3) se réécrit comme ceci :

$$\mathbf{E} = \left[\lambda^2_{flow}\mathbf{E}_{flow} + \lambda^2_{3D}\mathbf{E}_{3D} + \lambda^2_{2D}\mathbf{E}_{2D} + \lambda^2_{smooth}\mathbf{E}_{smooth}\right]\,, \tag{8}$$

où les paramètres lambdas sont des valeurs scalaires servant à pondérer l'influence des différents termes. Minimiser cette équation peut se formuler de la manière suivante :

$$\begin{aligned} \arg\min_{V^t} \;& \lambda^2_{flow}\delta_{\overline{F^t}}\|\nabla I^t.\left[J_\Pi V^t\right] + \frac{\mathrm{d}I^t}{\mathrm{d}t}\|^2 \\ &+ \lambda^2_{3D}\delta_{\mathcal{S}^t_m}\|V^t - V^t_f\|^2 \\ &+ \lambda^2_{2D}\delta_{\Omega^t_s} J_{\mathbf{\Pi}}\left[V^t - V^t_s\right] \\ &+ \lambda^2_{smooth}\|\nabla V^t\|^2\,, \end{aligned} \tag{9}$$

où δ est le symbole de Kronecker indiquant que ce terme ne s'applique qu'à un sous ensemble de points et $\overline{F^t}$ indique les points de la surface pour lesquels aucune information venant des points d'intérêt, 2D ou 3D, n'est disponible.

En dérivant l'équation (9), nous obtenons pour chaque point $\mathbf{P}$ de la surface, l'équation d'Euler-Lagrange discrète, de la forme :

$$\mathbf{A_P}V_\mathbf{P} + \mathbf{b_P} - \Delta V_\mathbf{P} = 0\,, \tag{10}$$

où Δ est l'opérateur de Laplace-Beltrami normalisé sur la surface.

4.1. Système linéaire

Étant donné que l'équation (10) met en jeu un ensemble de contraintes linéaires pour chaque point 3D de la surface, une solution est donnée par la résolution du système linéaire suivant :

$$\begin{bmatrix}\mathbf{L}\\ \mathbf{A}\end{bmatrix} V^t + \begin{bmatrix}\mathbf{0}\\ \mathbf{b}\end{bmatrix} = \mathbf{0}\,, \tag{11}$$

où $\mathbf{L}$ est la matrice laplacienne du maillage de la surface construite de telle manière que $\mathbf{L}(i,j)$ pondère la relation entre les points i et j (les poids du Laplacien sont discutés dans la section 5.1). $\mathbf{A}$ et $\mathbf{b}$ stockent toutes les contraintes visuelles sur le déplacement venant des termes d'attache aux données. Ce système linéaire est creux et peut être résolu en utilisant un solveur adapté tel que *Taucs* (Toledo *et al.*, 2001).

Il est intéressant de remarquer que cette formulation revisite le principe de déformation laplacienne de maillages de manière aussi rigide que possible (*as rigide as possible*) présenté dans la communauté du graphisme (Sorkine, Alexa, 2007). Bien que le schéma de déformation soit similaire, la différence se trouve dans les contraintes utilisées : des points ancre dans (Sorkine, Alexa, 2007) et des contraintes visuelles dans notre cas. Dans les deux cas, il est clairement identifié que le modèle de déformation ne prend pas en compte les rotations de manière explicite. Bien que cela présente un désavantage certain dans le cas où l'on dispose d'un faible nombre de contraintes, comme c'est souvent le cas dans les applications du graphisme, la densité des contraintes que nous utilisons permet de retrouver ces rotations sans recourir à une résolution non linéaire.

L'équation (9) peut aussi être résolue de manière itérative en appliquant la méthode de Jacobi. De cette manière on résout le système indépendamment en chaque point en utilisant la solution courante du voisinage comme présenté dans la section suivante.

4.2. Résolution itérative

Inspiré par les travaux de Horn et Schunck (Horn, Schunck, 1981), nous dérivons de l'équation (10) la résolution itérative suivante en chaque point de la scène :

$$\begin{aligned} v_x^{k+1} &= \overline{v}_x^k + A_x^x v_x^k + A_y^x v_y^k + A_z^x v_z^k - b_x \\ v_y^{k+1} &= \overline{v}_y^k + A_x^y v_x^k + A_y^y v_y^k + A_z^y v_z^k - b_y \\ v_z^{k+1} &= \overline{v}_z^k + A_x^z v_x^k + A_y^z v_y^k + A_z^z v_z^k - b_z \end{aligned} \tag{12}$$

où (v_x, v_y, v_z) et $(\overline{v}_x, \overline{v}_y, \overline{v}_z)$ représentent respectivement le déplacement propre et le déplacement moyen localement d'un point, l'indice k représente l'itération courante et les A_i^j et b_i sont les éléments de la matrice $\mathbf{A}$ et du vecteur $\mathbf{b}$ de l'équation (10).

Nous pouvons remarquer que les équations (12) sont indépendantes, à une itération donnée, pour chaque point de la surface. Ainsi l'implémentation de la résolution peut être massivement parallélisée. Dans ces équations, le déplacement local moyen pour un point 3D est donné par le voisinage de ce point en respectant la connectivité de la surface discrétisée et est pondéré exponentiellement en utilisant la taille des arêtes. Ainsi nous renforçons la relation qui lie deux points proches tout en empêchant les points aux frontières des objets d'être perturbés par des points lointains.

Cette formulation permet une approximation rapide du champ de déplacement. La rapidité et la précision de la résolution dépend fortement d'une bonne initialisation. Comme mentionné dans (Horn, Schunck, 1981), une bonne solution initiale peut être donnée par l'estimation obtenue à la trame précédente.

5. Détails d'implémentation

Cette section présente en détail les choix importants faits au moment de l'implémentation. Premièrement, nous discutons les poids qui déterminent, lors de la régula-

risation, l'influence du voisinage de la surface. Ensuite, nous présentons comment les grands et petits déplacements sont gérés séparément par le biais d'un algorithme en deux passes.

5.1. Poids laplaciens

Dans le terme de lissage de l'équation (9), l'opérateur de Laplace-Beltrami ∇^2, défini sur la surface de manière continue, est approché par la matrice Laplacienne du graphe du maillage $\mathbf{L}$, c'est-à-dire $\nabla^2 V^t = \mathbf{L}V^t$, où :

$$\mathbf{L}(i,j) = \begin{cases} \deg(P_i) & \text{si } i = j, \\ -w_{ij} & \text{si } i \neq j \text{ et } P_i \text{ est adjacent à } P_j, \\ 0 & \text{sinon,} \end{cases}$$

où les w_{ij} correspondent aux poids des arêtes et $\deg(P_i) = \sum_{j \neq i} w_{ij}$. La matrice $\mathbf{L}$ peut être purement combinatoire, c'est-à-dire $w_{ij} \in \{0, 1\}$, ou contenir des poids $w_{ij} \geq 0$.

5.1.1. Dans le cas de maillages watertight

Dans le cas où nous disposons d'une surface complète, c'est-à-dire, close et sans trou. Nous pouvons pré-traiter ces données pour obtenir des maillages réguliers. Typiquement, les maillages issus d'algorithme de reconstruction type enveloppe visuelle présentent souvent de très grandes disparités dans la taille de leurs arêtes et une faible homogénéïté dans la répartition spatiale de leur sommets. Une simple étape de ré-échantillonnage permet d'obtenir un maillage bien plus propre pour les traitements suivants sans pour autant altérer les propriétés de forme du maillage initial. Lorsque l'échantillonnage du maillage est uniforme, les poids cotangents, souvent utilisés en graphisme (Wardetzky *et al.*, 2007), permettent de garantir que la déformation appliquée à la surface maillée conservera au mieux les rigidités locales de la surface (Sorkine, Alexa, 2007).

5.1.2. Dans le cas des nuages de points

Dans le cas des nuages de points, la connectivité du maillage vient de celle de l'image de profondeur, c'est-à-dire que les points voisins dans l'image sont reliés sur la surface maillée par une arête. Cela aboutit à la construction d'un maillage cohérent, c'est-à-dire sans auto-intersection, mais avec potentiellement des arêtes de très grande taille correspondant aux discontinuités de la carte de profondeur. Pour gérer correctement ces discontinuités lors de la régularisation, nous proposons l'utilisation des poids suivants :

$$w_{ij} = -G(|P_i - P_j|, \sigma),$$

où G est un noyau gaussien, $|\cdot|$ est la distance euclidienne et σ l'écart type. En plus de fortement limiter la diffusion le long des grandes arêtes, les noyaux gaussiens sont aussi préconisés par Belkin *et al*. (Belkin *et al.*, 2008) pour leur propriété de convergence vers le cas continu de l'opérateur de Laplace-Beltrami lorsque la résolution du maillage augmente.

5.2. Algorithme en deux passes

Dans l'équation (9), les paramètres λ_{2D}, λ_{3D}, λ_{flow} et λ_d indiquent le poids, respectivement, des points d'intérêts 2D et 3D, du flot normal 2D et du laplacien. Une forte valeur indique une influence plus importante pour le terme associé.

Dans notre contexte, de manière similaire à (Xu *et al.*, 2010) en 2D, nous faisons confiance à nos points d'intérêts pour être robustes même lors de grands déplacements et nous sommes conscients que les contraintes de flot ne sont pas fiables quand la reprojection du déplacement est plus grande que quelques pixels dans les images. En conséquence, nous proposons une méthode itérative qui effectue deux minimisations successives de la fonctionnelle d'énergie avec deux jeux de paramètres différents. Les étapes de notre algorithme, illustré dans la figure 4, sont les suivantes :

1. Nous commençons par calculer les correspondances éparses 2D et 3D entre $\mathcal{S}^t$ et $\mathcal{S}^{t+1}$ et entre $\mathbf{I}^t$ et $\mathbf{I}^{t+1}$. Nous calculons également la matrice laplacienne $\mathbf{L}$ de notre surface discrétisée.

2. Nous résolvons l'équation (11), avec $\lambda_{flow} = 0$ et des valeurs plus importantes pour λ_{3D} et λ_{2D} que pour λ_{smooth}. Nous obtenons alors une première estimation de V^t dénotée V'^t qui récupère les grands déplacements de la surface.

3. Nous créons une surface déformée $\mathcal{S}'^t = \mathcal{S}^t + V'^t$ que nous projetons dans toutes les caméras en utilisant l'information de texture d'origine, venant de la projection de $\mathcal{I}^t$ sur $\mathcal{S}^t$. Nous obtenons alors un nouveau jeu d'images $\mathcal{I}'^t$.

4. Nous calculons alors les zonnes dans les images où la surface S'^t est visible ainsi que les contraintes denses de flot normal entre $\mathcal{I}'^t$ et $\mathcal{I}^{t+1}$ pour chaque point visible de la surface. Nous obtenons donc plusieurs contraintes par points échantillonnés sur la surface.

5. Tout comme dans l'étape 2, nous résolvons l'équation (11) en utilisant le flot calculé dans l'étape 4 et les points d'intérêts 2D et 3D calculés précédemment dans l'étape 1. Ces derniers sont utilisés comme des points d'ancrage ayant une contrainte de déplacement nul. Pour cette étape, nous utilisons des valeurs fortes de λ_{3D} et λ_{2D} et des valeurs plus faibles pour λ_{flow} et λ_{smooth}. Nous obtenons alors le déplacement entre $\mathcal{S}'^t$ et $\mathcal{S}^{t+1}$ dénoté V''^t et donc également une version raffinée de $V^t = V'^t + V''^t$. Cette seconde minimisation permet de récupérer de plus petits déplacements, mieux contraints par les contraintes de flot.

Nous avons observé par nos résultats que, dans la pratique, notre approche peut gérer aussi bien de grands déplacements que des petits. Ceci grâce aux points d'intérêts qui gèrent bien les grands déplacements et au flot de normal qui récupère mieux les détails précis.

6. Evaluations pour les maillages watertight

Pour notre évaluation nous avons utilisé aussi bien des données synthétiques que des données réelles :

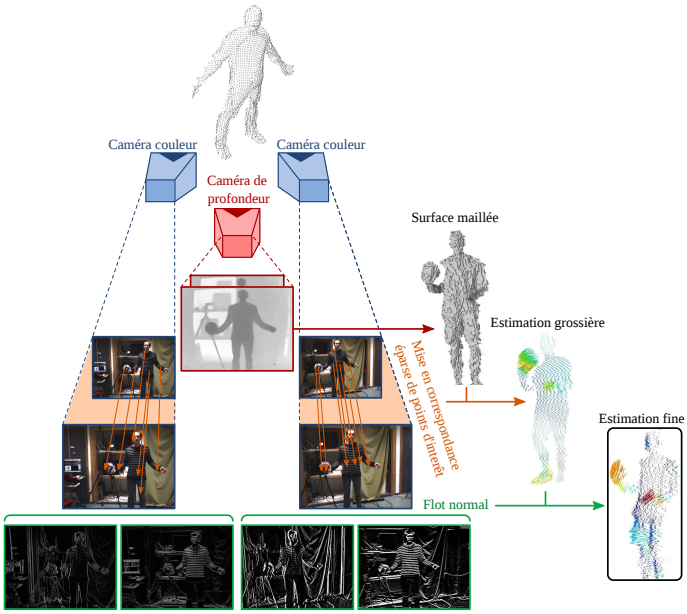


FIGURE 4 – Illustration des deux passes de notre algorithme dans le cas de deux caméras couleur et d'une caméra de profondeur.

1. Les données synthétiques ont été obtenues grâce à un modèle humain articulé, déformé au cours du temps pour créer une séquence de danse. Nous avons calculé le rendu de cette séquence dans dix caméras virtuelles de résolution 1 MPixels, réparties sur une sphère autour de la danseuse. Le modèle utilisé est un maillage triangulaire avec $7K$ sommets, déformé pour générer une séquence de 200 trames.

2. Les données réelles ont été récupérées à partir de banques de données accessibles au public. La première séquence a été prise à partir de 32 caméras 2 MPixels. Les maillages, obtenus avec EPVH, comportent $\sim 10K$ sommets. Nous avons également utilisé la séquence du *flashkick* de la base de données multi-vidéo *SurfCap* (Starck, Hilton, 2007b) de l'Université de Surrey. Cette séquence a été enregistrée à partir de huit caméras 2 MPixels, et produit des maillages lisses de $\sim 140K$ sommets.

6.1. Évaluation quantitative sur des données synthétiques

Grâce à l'algorithme décrit dans la section 5.2, nous avons calculé les champs de mouvement sur la séquence synthétique de la danseuse. Les figures 5a, 5b et 5c montrent le champ de déplacement sur une des trames de la séquence. Les flèches rouges désignent les contraintes issues des points d'intérêts 3D et de la projection des points d'intérêts 2D, alors que les bleues désignent les vecteurs du champ de déplacement dense 3D.

(a) (b) (c)

FIGURE 5 – Champ de déplacement sur plusieurs trames de notre séquence synthétique de danseuse (a), (b) et (c).

Comme les maillages sont cohérents dans le temps, nous avons pu obtenir la réalité terrain et donc évaluer nos résultats quantitativement. La figure 6 montre l'erreur sur l'angle et la taille de chaque vecteur de mouvement après chacune des deux étapes de régularisation de notre algorithme. Nous pouvons voir les avantages de l'utilisation des contraintes de flot normal pour affiner le champ de déplacement.

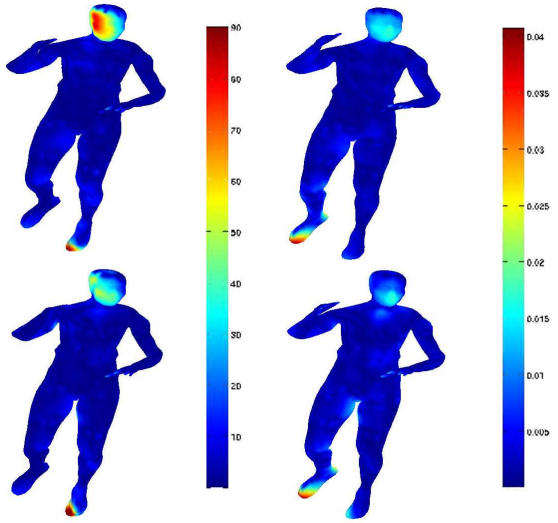
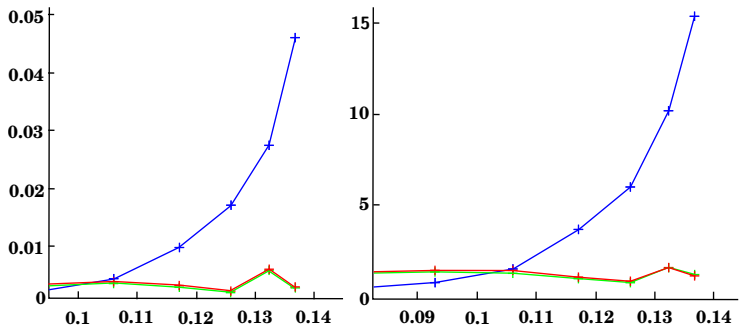


FIGURE 6 – Erreur sur le champ de déplacement : en angle en degré (gauche) et en norme en mètre (droite), après la première (haut) et la deuxième (bas) régularisation.

Les graphes de la figure 7 montrent des résultats quantitatifs sur deux séquences de synthèse. Chacune de ces séquences est composée de 34 flux de 15 trames montrant une sphère en mouvement.

Dans la première séquence la sphère subit un mouvement de translation pure et dans la seconde, le mouvement est une rotation par rapport au centre de la sphère. Dans les deux cas, l'intensité du mouvement est grandissante au fur et à mesure de la séquence générée ; avec par exemple jusqu'à 12°de rotation entre deux trames successives. Nous pouvons observer dans les graphes que la seconde passe de régularisation (en vert) permet d'obtenir grossièrement le même niveau d'amélioration des résultats par rapport à la première passe, en rouge ; et ce quel que soit l'amplitude du mouvement. Ceci est dû au fait que la première passe de notre méthode permet de retrouver les grands mouvements de telle manière que les déplacements résiduels se trouve à un niveau sous pixelique, niveau auquel l'information de flot devient cohérente et utilisable. Les graphes montrent aussi clairement que la qualité de nos résultas n'est pas dépendante de l'amplitude des mouvements, comme c'est le cas pour d'autre méthodes comme nous allons le voir plus loin.
Il est aussi intéressant de noter que l'apparence des courbes est strictement la même quel que soit le type de déplacement considéré : translation ou rotation. Pour cette raison, nous ne présentons que les résultats quantitatifs sur la séquence en translation ici. Cela signifie que, bien que notre modèle de déformation ne prenne pas directement en compte les rotations comme mentioné dans la section 4.1, nous sommes tout de même en mesure d'estimer correctement le mouvement de la scène.



(a) Erreur sur la norme (b) Erreur sur l'orientation

FIGURE 7 – (a) Norme (en mètres) et (b) angle (en degrés) d'erreur du déplacement éstimé en fonction de l'amplitude du mouvement réel de la surface (en mètres). En bleu : Vedula *et al.*, en rouge : la méthode proposée après la première régularisation, et en vert : après la seconde régularisation.

6.2. Comparaison

Dans le but de fournir une comparaison de notre méthode avec l'état de l'art, nous avons implémenté l'approche proposée par Vedula *et al.* dans (Vedula *et al.*, 2005). Puisque cet article présente trois différentes approches pour calculer le flot de scène, nous avons choisi d'utiliser celle qui utilise les mêmes données en entrée que la nôtre,

à savoir "*Multiple cameras, known Geometry*". Nous avons pour ce faire utilisé la denière implémentation OpenCV du calcul du flot optique à l'aide de l'algorithme de Lukas-Kanade (Lucas, Kanade, 1981) avec les paramètres standards. Cette information de flot optique est ensuite intégrée comme décrit dans l'article pour en déduire le flot de scène. Les graphes de la figure 7 présentent les niveaux d'erreur obtenus utilisant la méthode de Vedula *et al*. (courbe bleue) en comparaison avec les résultats de notre méthode. Les séquences utilisées pour faire nos comparaisons sont les deux séquences décrites en fin de section précédente. Comme prévu, notre méthode présente des niveaux d'erreur bien plus convaincants dès que l'amplitude du mouvement dépasse la taille des pixels dans les images. Notre hypothèse pour expliquer ceci est que pour des déplacements sub-pixeliques, le calcul du flot optique peut-être très précis et ainsi il fournit une meilleure information que celle apportée par le flot normal uniquement.

Il est intéressant de noter que nos résultats sont fortement corrélés avec la résolution du modèle géométrique utilisé, c'est-à-dire la densité de sommets du maillage. Tandis que Vedula *et al*. font de la régularisation dans l'espace image, nous la faisons directement sur la surface. Ainsi nous pourrions augmenter légèrement la qualité de nos résultats en augmentant la résolution des maillages utilisés.

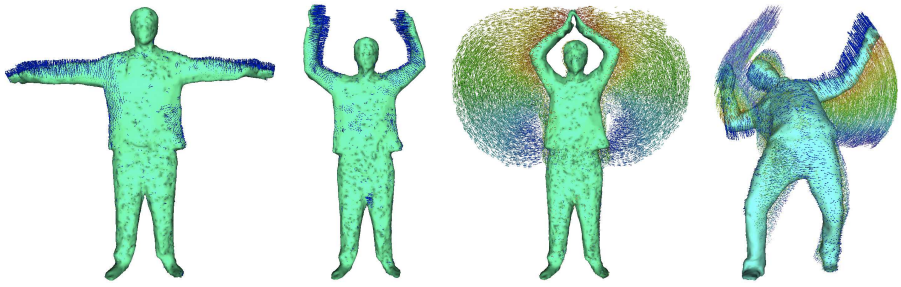
6.3. Expériences sur des données réelles

Notre première séquence réelle montre un sujet qui réalise des actions simples : il déplace ses deux mains à partir des hanches jusqu'au dessus de sa tête. Le sujet porte des vêtements amples et bien texturés ce qui permet de calculer un nombre élevé et fiable de correspondances 2D et 3D.

Les figures 8a, 8b et 8c montrent le mouvement instantané récupéré en utilisant notre méthode. Notez que nous ne calculons qu'un mouvement dense sur la surface et non une déformation du maillage. Ainsi, nous n'avons pas une connectivité constante dans le temps et nous ne pouvons pas effectuer le suivi des sommets du maillage sur toute la séquence. Par conséquent, l'évaluation quantitative des données n'est pas possible, mais la visualisation des résultats est très satisfaisante. La figure 8d montre le champ de déplacement accumulé sur toute la séquence.

Nous avons également calculé le champ de déplacement 3D sur la séquence du *flashkick* qui est très populaire. Dans cette séquence difficile, le sujet porte des vêtements amples avec peu d'information de texture. En outre, l'amplitude du mouvement est très élevée entre deux trames. Nous pouvons donc calculer moins de correspondances 2D et 3D. Elles sont pourtant nécessaires pour récupérer les grands déplacements.

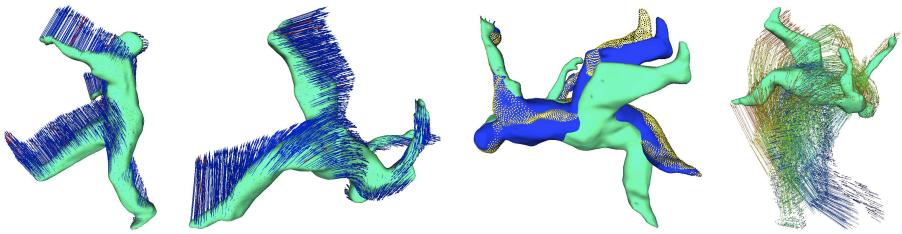
Nous avons cependant réussi à calculer un champ de mouvement cohérent sur la plupart des trames (voir les figures 9a et 9b). Sur certaines trames, notre algorithme n'a pas trouvé de points d'intérêts sur les jambes ou les pieds du danseur, le champ de mouvement calculé à partir de ces indices montre bien la bonne direction, mais pas la bonne norme des vecteurs. Le manque de contraintes visuelles pour la première



(a) (b) (c) (d)

FIGURE 8 – (a) et (b) : champs de déplacement sur certaines trames de nos données réelles. (c) et (d) : historiques du mouvement accumulés sur toute la séquence (les couleurs indiquent l'ancienneté du mouvement).

estimation du champ de mouvement ne permet pas de calculer certains déplacements complètement, le déplacement restant ne peut pas être récupéré entièrement avec les contraintes de flot normal. La figure 9c montre une trame problématique où le mouvement de la jambe droite du danseur n'est pas correctement calculé. Pour visualiser cette erreur, nous avons affiché les surfaces d'entrée au temps t et $t+1$ (respectivement cyan et bleu foncé), tandis que la surface déplacée avec le champ de mouvement calculé est indiquée par des points jaunes. Enfin, la figure 9d montre l'historique du mouvement sur quelques trames.



(a) (b) (c) (d)

FIGURE 9 – Champs de déplacement sur plusieurs trames de la séquence du flashkick (a) et (b), mouvement partiellement calculé (c), et historique du mouvement sur toute la séquence (d) (les couleurs indiquent l'ancienneté).

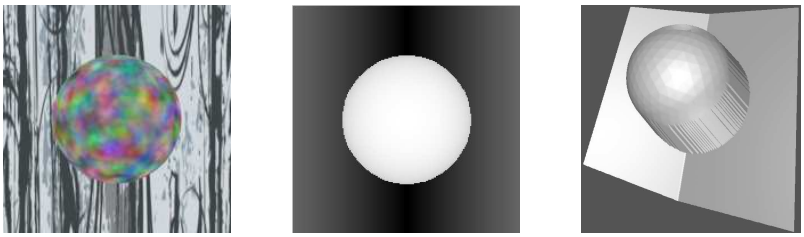
7. Evaluations pour les cartes de profondeur

Pour procéder à l'évaluation de la méthode proposée, nous avons utilisé plusieurs séquences dans différentes conditions. Pour commencer, nous avons créé des données de synthèse pour permettre une évaluation quantitative. Dans un second temps nous avons procédé à l'acquisition et au traitement de données réelles à l'aide de deux confi-

gurations différentes, avec une ou plusieurs caméras couleur. Les différentes configurations ainsi que les résultats obtenus sont détaillés dans cette section.

7.1. Données de synthèse

Les données de synthèse représentent une sphère en mouvement devant deux plans également en déplacement. Cette scène est ensuite projetée dans deux caméras de synthèse de 1M pixels. Nous utilisons le *depth buffer* d'une de ces deux caméras virtuelles pour obtenir la carte de profondeur de la scène (voir figure 10a et 10b). Cette carte de profondeur a été rééchantillonnée à une résolution de 200×200 et utilisée pour créer un maillage (voir figure 10c). Dans la séquence générée, la sphère se déplace en s'éloignant de la caméra, tandis que l'un des plans se déplace vers le haut et l'autre vers le bas. Il est important de noter que l'extension de la méthode proposée à $N > 1$ caméras couleur ne change rien à la formulation, cela ne fait qu'empiler plus de contraintes dans l'équation (11).



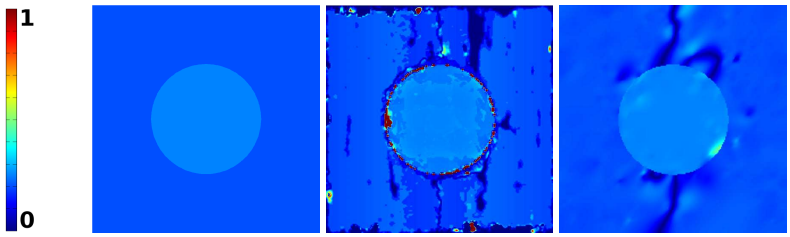
(a) Image couleur (b) Carte de profondeur (c) Maillage associé

FIGURE 10 – Données de synthèse : image couleur (a), carte de profondeur (b) et le maillage inféré (c).

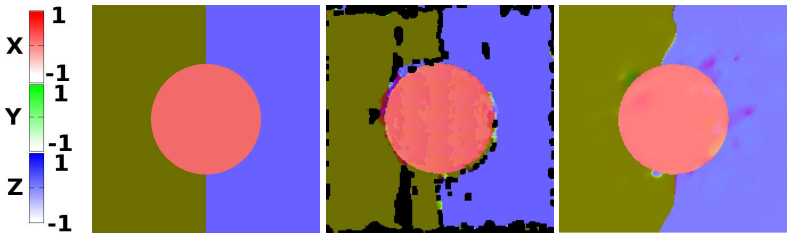
Nous avons comparé notre approche à la méthode de référence présentée dans (Vedula *et al.*, 2005) dans le cas "*Single camera, known scene geometry*". Cette méthode requiert les mêmes données en entrée que la nôtre et est aussi extensible au cas multi-caméra.

Les résultats obtenus sont présentés dans la figure 11 où les normes et orientations des déplacements 3D sont représentés depuis le point de vue de la caméra avec un code couleur. La figure 12 montre l'erreur en chaque point de la surface maillée, tandis que la table 1 présente une comparaison numérique.

Les résultats obtenus montrent que la méthode proposée est capable de gérer correctement les discontinuités de la carte de profondeur entre la sphère et les plans. Néanmoins, à l'endroit où les deux plans se croisent, il y a une ambiguïté qui conduit à supposer que les deux plans sont connectés sur la surface maillée. Ainsi la régularisation a du mal à évaluer correctement les déplacements dans cette zone. Ce comportement était attendu dans la mesure où notre hypothèse de régularisation en 3D est violée à la jonction des deux plans. C'est-à-dire que les deux plans se touchent

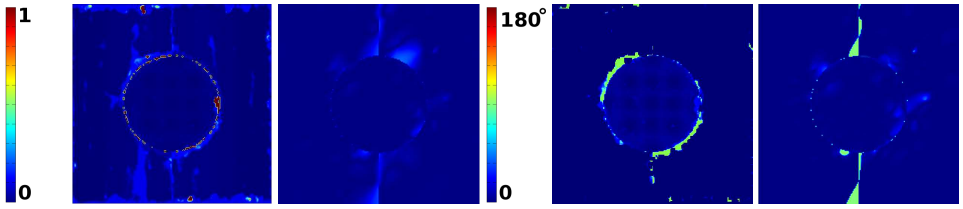


(a) Norme du déplacement 3D (code couleur)



(b) Direction du déplacement 3D (code couleur)

FIGURE 11 – Résultats sur des données de synthèse et comparaison entre vérité terrain (gauche), la méthode de Vedula (Vedula *et al.*, 2005) (centre) et notre méthode (droite).



(a) Erreur sur la norme du déplacement 3D (b) Erreur sur l'orientation du déplacement 3D

FIGURE 12 – Erreur sur des données de synthèse avec comparaison entre la méthode de Vedula (Vedula *et al.*, 2005) (gauche) et la méthode proposée (droite).

mais ont des déplacements complétement différents. Cet exemple met ainsi en avant la force et la faiblesse de notre méthode.

Tableau 1 – Erreurs numériques sur des données de synthèse avec comparaison entre la méthode de Vedula et la méthode proposée.

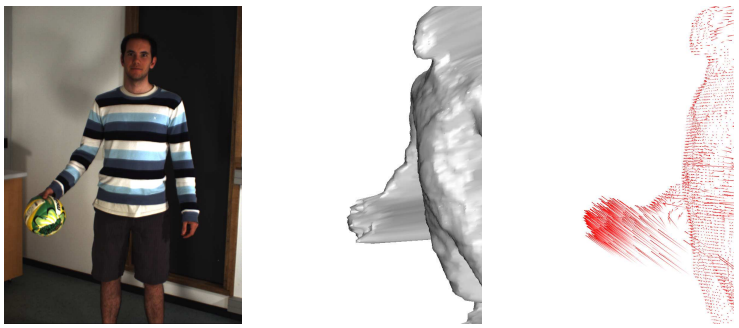
	Vedula (Vedula *et al.*, 2005)		Notre méthode	
Erreur	Moyenne	Médiane	Moyenne	Médiane
Norme	33%	7.27%	**8.68%**	**2.33%**
Angle	8.6°	**0.10°**	**2.7°**	0.12°

Les expérimentations menées montrent qu'avec de bonnes textures et des données de synthèse, les contraintes du flot normal n'aident pas réellement à améliorer les résultats puisque les déformations sont strictement rigides et beaucoup de points d'intérêt sont détectés et appariés correctement, ce qui suffit à retrouver les mouvements dans le cas de scènes basiques. L'ajout de caméras couleur supplémentaires n'améliore pas significativement l'estimation des déplacements puisque les points d'intérêt détectés tendent à être les mêmes d'une image à l'autre dans le cas d'un faible parallaxe.

7.2. Données réelles

Nous avons aussi procédé à des expérimentations sur des données réelles acquises avec deux systèmes différents : (1) un système multi-caméra composé d'une caméra temps de vol Swiss Ranger SR4000 de résolution 176×144 accompagnée de deux caméras couleur de 2M pixels, et (2) une caméra Kinect de Microsoft capable de fournir un flux d'images couleur, chacune alignée sur une carte de profondeur de résolution 640×480. Le système multi-caméra avec la caméra temps de vol a été calibré en utilisant les travaux présentés dans (Hansard *et al.*, 2011).

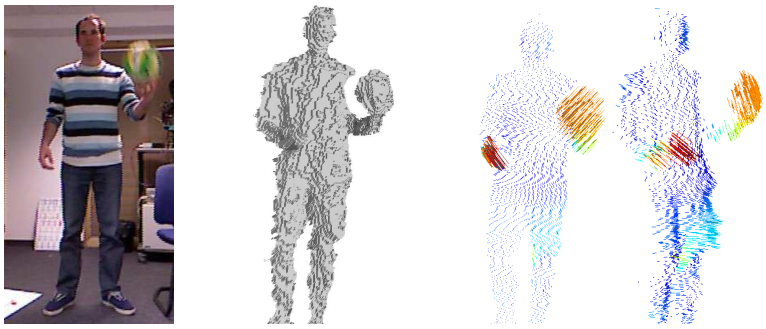
Pour tester efficacement notre méthode nous avons acquis avec les deux systèmes une scène identique dans laquelle un homme se tient debout dans une pièce et joue avec une balle, la faisant sauter d'une main à l'autre. Cette scène présente à la fois de grandes discontinuités dans la carte de profondeur et des déplacements larges et rapides.



(a) Image couleur (b) Maillage (c) Déplacements

FIGURE 13 – Données en entrée : une des deux images couleur (gauche) et la surface calculée (centre). Résultat : le champ de déplacement 3D obtenu sur les données de la caméra temps de vol (droite), la couleur encode l'orientation des vecteurs : départ en blanc, arrivée en rouge.

Les résultats présentés sur les figures 13 et 14 démontrent l'intérêt ainsi que la faisabilité de notre méthode sur des données réelles. Les codes couleurs des figures 13c et 14c indiquent respectivement l'orientation et l'intensité du déplacement. Le champ



(a) Image couleur (b) Maillage (c) Champs de déplacements

FIGURE 14 – Données en entrée : l'image couleur (gauche) et la surface calculée (centre). Résultat : le champ de déplacement 3D obtenu sur les données de la caméra Kinect (droite), la couleur encode la norme des vecteurs.

de déplacement estimé est cohérent avec les actions exécutées par la personne. L'utilisation de deux caméras couleur dans le cas de la caméra temps de vol permet d'obtenir un résultat satisfaisant bien que les données géométriques soient très bruitées. En ce qui concerne la caméra Kinect, la résolution des données acquises induit un maillage de haute densité qui accentue la complexité du système linéaire. Dans ce cas, une implémentation parallèle peut permettre de balancer la complexité des données.

8. Conclusion et discussion

La contribution de ce travail est double : premièrement, nous avons présenté une méthode unifiée qui permet de combiner des informations photométriques pour calculer le mouvement d'une surface, d'autre part, nous avons introduit une méthode itérative qui permet de gérer de grands déplacements tout en récupérant les petits détails. Comme le montrent les résultats, notre méthode est assez robuste et polyvalente. En effet, elle peut s'adapter sans surcoût pour l'utilisateur sur des systèmes multi-caméra de nature différente. Nos expériences vont dans ce sens en démontrant l'adaptabilité de la méthode présentée à des systèmes contenant de une à 32 caméras couleur, avec ou sans capteur de profondeur. Néanmoins, nos expériences ont mis en évidence certaines faiblesses potentielles.

Comme nous le pensions, s'appuyer sur des caractéristiques visuelles impose d'avoir une bonne information de texture dans les images. Notre méthode pourrait être améliorée par l'ajout d'autres contraintes, par exemple, un critère de cohérence photométrique tel que celui utilisé par Pons *et al.* dans (Pons *et al.*, 2005). Il serait interessant d'étudier une meilleure manière d'intégrer l'information de flot de normal. Pour le moment, nous n'utilisons cette information que pour retrouver les détails du champ de déplacements. Par la mise en oeuvre d'une approche multi-échelle, nous pourrions intégrer cette contrainte même dans le cas de déplacements conséquents. Nous avons

rejeté les approches multi-échelle qui proposent un lissage dans l'espace image sous prétexte qu'elles souffrent de sur-lissage pour les pixels la frontière des objets. Néanmoins, puisque nous dispons de la géométrie de la scène, il est possible d'imaginer effectuer un lissage dans l'espace image qui tient compte de la géométrie de la scène.

Nous avons aussi mis en avant dans nos expérimentations (section 7.1) un cas limite pour notre méthode. Si deux objets distincts et aux déplacements différents sont proches au point d'être assimilés à la même forme dans la reconstruction de la scène ; alors notre hypothèse de régularisation est violée et la précision de nos résultats chute. Il s'agit d'un cas marginal mais qui présente tout de même une limitation en l'état actuel de nos travaux. Pour y remédier, il faudrait avoir accès à une meilleure information de géométrie. La méthode que nous proposons donne de toute façon des informations utiles et fiables sur les propriétés intrinsèques d'une séquence 4D. La connaissance du déplacement instantané peut être utilisée comme donnée d'entrée pour de nombreuses tâches en vision par ordinateur, telles que le suivi de surface, le transfert de mouvement ou la segmentation de maillages.

Même si nous n'avons pas mis l'accent sur les performances de calcul pour notre première implémentation, nous sommes certains que la plupart des calculs pourraient s'exécuter en parallèle. En effet, l'extraction des points d'intérêts 2D, ainsi que le calcul des contraintes de flot normal, sont indépendants par caméra. De plus, des implémentations temps-réel de SIFT et des méthodes de flux optique existent déjà. La propriété linéaire de la régularisation permet de s'attendre à une exécution en temps-réel également.

Ce travail a été partiellement financé par OSEO, l'agence française pour l'innovation, dans le cadre du programme de recherche QUAERO.

Bibliographie

Barron J.-L., Fleet D.-J., Beauchemin S. (1994). Performance of Optical Flow Techniques. *International Journal of Computer Vision*.

Basha T., Moses Y., Kiryati N. (2010). Mutli-View Scene Flow Estimation : A View Centered Variational Approach. In *Computer vision and pattern recognition.*

Bay H., Ess A., Tuytelaars T., Van Gool L. (2006). SURF : Speeded Up Robust Features. *Computer Vision and Image Understanding*.

Belkin M., Sun J., Wang Y. (2008). Discrete Laplace Operator on Meshed Surfaces. In *Proceedings of the symposium on computational geometry.*

Cagniart C., Boyer E., Ilic S. (2010). Probabilistic Deformable Surface Tracking From Multiple Videos. *European Conference on Computer Vision*.

Devernay F., Mateus D., Guilbert M. (2006). Multi-Camera Scene Flow by Tracking 3-D Points and Surfels. In *Computer vision and pattern recognition.*

Hansard M., Horaud R., Amat M., Lee S. (2011). Projective Alignment of Range and Parallax Data. In *Computer vision and pattern recognition.*

Harris C., Stephens M. (1988). A combined corner and edge detector. In *Alvey vision conference.*

Horn B., Schunck B. (1981). Determining Optical Flow. *Artificial Intelligence*.

Huguet F., Devernay F. (2007). A Variational Method for Scene Flow Estimation From Stereo Sequences. In *International conference on computer vision.*

Isard M., MacCormick J. (2006). Dense Motion and Disparity Estimation via Loopy Belief Propagation. In *Asian conference on computer vision.*

Letouzey A., Petit B., Boyer E. (2012). Flot de scène à partir d'images couleur et de cartes de profondeur. In *Actes de la conférence RFIA 2012.*

Li R., Sclaroff S. (2008). Multi-scale 3D Scene Flow from Binocular Stereo Sequences. *Computer Vision and Image Understanding*.

Liu C., Yuen J., Torralba A., Sivic J., Freeman W. (2008). SIFT Flow : Dense Correspondence across Different Scenes. In *European conference on computer vision.*

Lowe D. (2004). Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*.

Lucas B., Kanade T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *International joint conference on artificial intelligence.*

Naveed A., Theobalt C., Rossl C., Thurn S., Seidel H. (2008). Dense Correspondence Finding for Parametrization-free Animation Reconstruction from Video. In *Computer vision and pattern recognition.*

Neumann J., Aloimonos Y. (2002). Spatio-Temporal Stereo Using Multi-Resolution Subdivision Surfaces. *International Journal of Computer Vision*.

Pons J.-P., Keriven R., Faugeras O. (2005). Modelling Dynamic Scenes by Registering Multi-View Image Sequences. In *Computer vision and pattern recognition.*

Rabe C., Müller T., Wedel A., Franke U. (2010). Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time. In *European conference on computer vision.*

Sorkine O., Alexa M. (2007). As-Rigid-As-Possible Surface Modeling. In *Eurographics symposium on geometry processing.*

Starck J., Hilton A. (2007a). Correspondence Labeling for Wide-Timeframe Free-Form Surface Matching. In *European conference on computer vision.*

Starck J., Hilton A. (2007b). Surface Capture for Performance-Based Animation. *IEEE Computer Graphics and Applications*.

Toledo S., Chen D., Rotkin V. (2001). http://www.tau.ac.il/~stoledo/taucs/.

Varanasi K., Zaharescu A., Boyer E., Horaud R. P. (2008). Temporal Surface Tracking Using Mesh Evolution. In *European conference on computer vision.*

Vedula S., Baker S., Rander P., Collins R., Kanade T. (2005). Three-Dimensional Scene Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wardetzky M., Mathur S., Kälberer F., Grinspun E. (2007). Discrete Laplace Operators : No free lunch. In *Eurographics symposium on geometry processing.*

Wedel A., Rabe C., Vaudrey T., Brox T., Franke U., Cremeres D. (2008). Efficient Dense Scene Flow from Sparse or Dense Stereo Data. In *European conference on computer vision.*

Xu L., Jia J., Matsushita Y. (2010). Motion Detail Preserving Optical Flow Estimation. In *Computer vision and pattern recognition.*

Zaharescu A., Boyer E., Varanasi K., Horaud R. P. (2009). Surface Feature Detection and Description with Applications to Mesh Matching. In *Computer vision and pattern recognition.*

Zhang Y., Kambhamettu C. (2001). On 3D Scene Flow and Structure Estimation. In *Computer vision and pattern recognition.*